# Blind Signal Classification via Sparse Coding

Youngjune Gwon
MIT Lincoln Laboratory
gyj@ll.mit.edu

Siamak Dastangoo
MIT Lincoln Laboratory
sia@ll.mit.edu

Carl Fossa
MIT Lincoln Laboratory
cfossa@ll.mit.edu

H. T. Kung
Harvard University
kung@harvard.edu

***Abstract*—We propose a novel RF signal classification method based on sparse coding, a popular technique for image recognition in machine learning. We treat sparse coding as a configurable framework and employ a convolutional sparse coder that extracts the maximal similarity from samples of an unknown received signal against an overcomplete dictionary of matched filter templates. Such dictionary can be either generated or learned via unsupervised algorithms. Under this approach, we can achieve blind signal classification with no prior knowledge about signals (*e.g.*, MCS, pulse shaping) in an arbitrary RF channel. Since modulated RF signals undergo pulse shaping to aid the matched filter detection by a receiver for the same radio protocol, we can exploit variability in relative similarity against the dictionary atoms as the key discriminating factor to build our classifiers. We present empirical validation of the proposed classification method. Our results indicate that we can separate different classes of digitally modulated signals from blind sampling with 70.3% recall and 24.6% false alarm at 10 dB SNR. If a labeled dataset were available for supervised classifier training, we can enhance the classification accuracy to 87.8% recall and 14.1% false alarm.**

## I. Introduction

Cognitive radios have emerged over the last decade as a new means to share radio spectrum, the most expensive resource to build a wireless network. For commercial applications, Dynamic Spectrum Access (DSA) [1] presents a compelling model to improve the utility of radio spectrum resources. Much of contemporary research has viewed cognitive radios as the secondary user of a licensed channel and focused on developing the mechanism to opportunistically access the channel to its maximal spectral efficiency.

While commercial opportunities are promising, the applicability of cognitive radios for tactical networking seems even more adequate. The primary advocate for tactical cognitive radio systems is intelligent decision making that can enhance resiliency against a hostile, fiercely competing radio environment. There has been significant amount of previous work devoted to algorithmic approaches for a cognitive strategy layer, including game-theoretic frameworks [2]–[5] to sequential decision making [6]–[8].

These approaches have provided a strong foundation for cognitive tactical radio systems, yet their performance highly depends on the lower layer capability such as sensing, detection, and inference of radio signals. In order to operate the cognitive strategy layer, our claim is that we require intelligent sensing mechanisms enabled by learning. In this paper, we focus on the development of such mechanisms. Particularly, we use sparse coding [9], a feature learning technique widely used in machine learning, to perform blind and semi-supervised signal classification for cognitive radios.

Our methods are new and unconventional to the field of signal detection and estimation. Our methods can learn over time after bootstrapping with no prior knowledge about RF signals of interest and achieve a 72.6% recall for blind signal classification under a reasonably good SNR. If a labeled dataset were available for semi-supervised training, our classifiers would have achieved a 87.8% recall with 14.1% false alarms, all without any protocol-specific knowledge about modulation of radio signals.

The rest of the paper is organized as follows. In Section II, we provide a comprehensive background on sparse coding. In Section III, we describe a discriminative framework that employs sparse coding as the primary means to extract features from raw data in a powerful classification pipeline. Section IV presents our RF signal classification methods. We propose a method for blind signal classification before presenting a semi-supervised approach under the availability of a labeled dataset. We evaluate the proposed classification methods in Section V. In Section VI, we discuss related work, and Section VII concludes the paper.

## II. Sparse Coding Background

This section presents a background on sparse coding and dictionary learning.

### A. Sparse Coding

Sparse coding [9] is an unsupervised method to learn a dictionary of overcomplete bases that represent data efficiently. Each basis vector in the dictionary is also known as an *atom*. The mathematical objective of sparse coding is to describe an input vector as a *sparse* linear combination of the basis vectors from the dictionary.

Fig. 1 explains the sparse coding problem. Given an $N$-dimensional input $\mathbf{x} \in \mathbb{R}^N$ and dictionary $D \in \mathbb{R}^{N \times K}$, sparse coding seeks for a sparse representation $\mathbf{y} \in \mathbb{R}^K$ that minimizes the loss function

$$J(\mathbf{x}, D) = \min_{\mathbf{y} \in \mathbb{R}^K} \frac{1}{2} \|\mathbf{x} - D\mathbf{y}\|_2^2 + \lambda \psi(\mathbf{y}), \qquad (1)$$

where the first term optimizes the reconstructive error, and the second term is due to regularization to control sparsity of $\mathbf{y}$.

Distribution A. This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and rec-ommendations are those of the authors and are not necessarily endorsed by the United States Government.

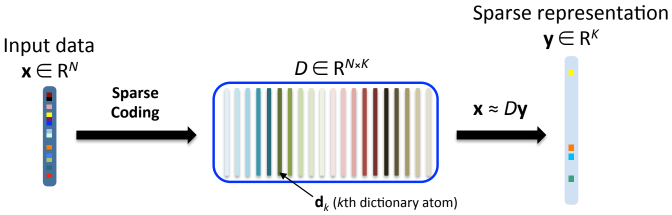
Input data
$\mathbf{x} \in \mathrm{R}^N$
Sparse Coding
$D \in \mathrm{R}^{N \times K}$
$\mathbf{x} \approx D\mathbf{y}$
Sparse representation
$\mathbf{y} \in \mathrm{R}^K$
$\mathbf{d}_k$ (kth dictionary atom)

Fig. 1. Sparse coding

Sparse coding considers $\psi(.)$ under two regularization strategies. First of all, we can adopt the $\ell_0$ penalty, *i.e.*, $\lambda \|\mathbf{y}\|_0$, to strictly regulate the total number of nonzero elements in $\mathbf{y}$. Although the $\ell_0$-norm of $\mathbf{y}$ gives a precise control, it is known to be NP-hard [10].

The second approach of regularization resorts to convex relaxation of the first. Instead of the computationally hard $\ell_0$-minimization, we can use the $\ell_1$ penalty term $\lambda \|\mathbf{y}\|_1$ instead. This is also known as *basis pursuit* [11] or the least absolute shrinkage selection operator (LASSO) [12]. It is well known that the $\ell_1$-minimization yields a sparse solution equivalent to its $\ell_0$ counterpart, but there is no analytic link between the value of $\lambda$ for $\ell_1$-norm and $\|\mathbf{y}\|_0$ [13].

*1) Orthogonal Matching Pursuit:* While exact determination via the $\ell_0$-minimization is hard, approximate solutions for optimizing $\ell_0$-norm are possible. Especially, fast greedy algorithms are possible by selecting the dictionary atoms sequentially from specifically enforcing sparsity requirement such as $S$-sparse $\mathbf{y}$:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \|\mathbf{x} - D\mathbf{y}\|_2^2 \quad \text{s.t.} \;\; \|\mathbf{y}\|_0 \le S. \tag{2}$$

Orthogonal Matching Pursuit (OMP) [14] selects the best dictionary atom by evaluating the inner product between the input and a dictionary atom and uses least squares to accurately settle the coefficients inside $\mathbf{y}$ iteratively for each round.

*2) Basis Pursuit:* The $\ell_1$-minimization for the sparse coding problem can be written as

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \|\mathbf{y}\|_1 \quad \text{s.t.} \;\; D\mathbf{y} = \mathbf{x}. \tag{3}$$

One approach for this optimization is linear programming [15]. Eq. (3), however, is not in the standard dual of a linear program

$$\min \mathbf{c}^\top \mathbf{y} \quad \text{s.t.} \;\; D\mathbf{y} = \mathbf{x}, \;\; \mathbf{y} \ge 0.$$

Chen, Donoho & Saunders [11] recommend make the following translations

$$\mathbf{y} \Leftrightarrow (\mathbf{u}, \mathbf{v}), \;\; \mathbf{c}^\top \Leftrightarrow (\mathbf{1}^\top, \mathbf{1}^\top), \;\; D \Leftrightarrow (D, -D).$$

Subsequently, solving

$$\min \mathbf{u} + \mathbf{v} \quad \text{s.t.} \;\; D\mathbf{u} - D\mathbf{v} = \mathbf{x}, \;\; \mathbf{u}, \mathbf{v} \ge 0$$

gives the $\ell_1$-minimization solution via linear programming.

### B. Dictionary Learning

How can we learn a dictionary for sparse coding? A dictionary is trained by an unsupervised learning algorithm such as K-means clustering. A classical approach [16] examines the projected first-order stochastic gradient descent in a sequence of updates for $D$

$$D_t = \Pi_{\mathcal{C}} \left[ D_{t-1} - \frac{\rho}{t} \nabla J(\mathbf{x}_t, D_{t-1}) \right], \tag{4}$$

where $\rho$ is the gradient step, $\Pi_{\mathcal{C}}$ is the orthogonal projector on $\mathcal{C}$, and unlabeled training examples $\{\mathbf{x}_k\}_{k=1}^T$.

In principal component analysis (PCA), we learn a complete set of basis vectors—*i.e.*, a square matrix of eigenvectors. Dictionary learning for sparse coding aims to learn an overcomplete set of basis vectors such that the column dimension of $D$ is larger than its row dimension. (Recall $D \in \mathbb{R}^{N \times K}$, so $K \ll N$.) The advantage of having an overcomplete bases is that we can better capture structures and patterns inherent in the input data more conveniently.

K-SVD [17] is a fast iterative algorithm for PCA-like basis learning. The inner loop of K-SVD has two phases. First, it performs batch sparse coding with current dictionary. Using the notation $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_T]$, the batch sparse coding yield the corresponding matrix of sparse codes $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_T]$ such that $\mathbf{X} \approx D\mathbf{Y}$. In the next phase, K-SVD updates each dictionary atom in $D$ by rank-1 update via singular value decomposition of residual matrix for the atom. K-SVD also updates each sparse code in $\mathbf{Y}$ accordingly. The K-SVD optimization is given by

$$\min_{D, \mathbf{Y}} \|\mathbf{X} - D\mathbf{Y}\|_F^2 \qquad \text{s.t.} \;\; \|\mathbf{y}_k\|_0 \le S \;\; \forall k. \tag{5}$$

Because of the batch sparse coding phase, K-SVD requires a sparse coder. We can use OMP, $\ell_1$-minimization via linear programming or LASSO.

## III. Discriminative Sparse Coding Framework

In this section, we present a discriminative sparse coding framework to build a high-performance classification pipeline. We explain unsupervised feature learning method based on sparse coding and dictionary training. Given the learned feature mapping, we describe how we can perform feature extraction, train classifiers, and predict a class label.

### A. Unsupervised Feature Learning via Sparse Coding

Typically, an unsupervised method is used to learn a feature representation of raw data. Since the feature mapping should be generally applicable and descriptive of all classes of data, feature learning takes in randomly mixed, unlabeled training examples. Sparse coding and dictionary training provide an unsupervised feature learning algorithm that consists of the following steps as illustrated in Fig. 2:

1) Form input patches $\mathbf{x}$ from measured/received signal data that are unlabeled of their classes;
2) (Optionally) apply preprocessing such as normalization and whitening;

3) Learn a feature-mapping via joint sparse coding (compute $\mathbf{y}$) and dictionary ($D$) training.

In summary, unsupervised feature learning takes the unlabeled dataset $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_T]$ of random input patches (each $\mathbf{x}_i$ with dimension $N$), undergoes sparse coding and dictionary learning, and yield a function $f_{\text{ext}} : \mathbb{R}^N \mapsto \mathbb{R}^K$. The transformation via $f_{\text{ext}}$ converts the raw data input $\mathbf{x}$ to sparse code $\mathbf{y}$ in the feature space learned by sparse coding and dictionary training. For classification, we use the sparse code $\mathbf{y}$ as a feature vector whose $K$ elements are features of the input $\mathbf{x}$ according to dictionary $D$.

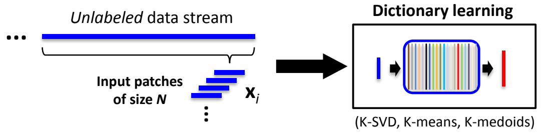

Fig. 2. Unsupervised learning via sparse coding and dictionary training

## B. Supervised Classifier Training

A representational feature mapping learned from the unsupervised method plays a crucial role for classification tasks. Having the feature mapping alone, however, is usually insufficient to classify data. Classifiers take a feature vector as the input, and they should be instructed with the ground truth class (*i.e.*, supervision) about the feature inputted. Therefore, classifier training is typically done by a supervised method such as logistic regression [18] and support vector machine (SVM) [19]. Supervised classifier training is depicted in Fig. 3. Note the labeled input $\{\mathbf{x}_i, l_i\}$, where $l_i$ designates the class label for an input $\mathbf{x}_i$.

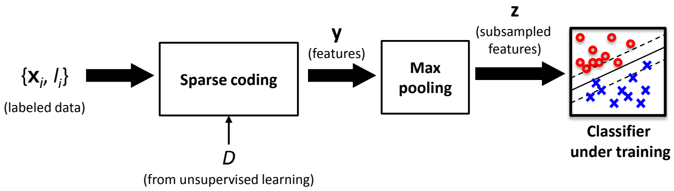

Fig. 3. Supervised classifier training

Under the context of tactical networking scenarios, it may be too optimistic to assume the availability of labeled dataset for supervised classifier training. This is because of the null *a priori* assumption for an adversarial radio network. Signal examples of the adversary could be hard to acquire for pre-analysis before a field operation. However, we can assume a plenty of signal examples for the friendly network. With only friendly network signal examples, one can employ one-class classifier [20] instead.

## C. Subsampling Features with Max Pooling

In Fig. 3, feature vectors (*i.e.*, sparse code $\mathbf{y}$) go through one more processing step known as *max pooling* before being inputted to a classifier under training. If all feature vectors resulted from a stream of input vectors were used straightforwardly for classification, we could overwhelm the classifier training. The dimensionality of feature vectors is highly correlated with the complexity of classifiers. Usually, a complex classification model leads to classifier overfit, which is the discrepancy in the classification results between the training and test datasets. It is therefore customary to reduce the number of extracted features by subsampling.

Pooling, popular in convolutional neural networks [21], operates over multiple (sparse) feature representations and aggregates to a higher level of features in reduced dimension. Pooling is by no means to discard any useful information. An important property of the pooled feature representation is translation invariance. Max pooling [22] takes the maximum value for the elements in the same position over a group of feature vectors. For example, consider max pooling of $L$ sparse codes $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L\}$ that yields the pooled feature vector $\mathbf{z}$ as in Fig. 4. Noting $\mathbf{y}_k = [y_{k,1} \ \ldots \ y_{k,K}]$ and $\mathbf{z} = [z_1 \ \ldots \ z_K]$, max pooling operation is given by $z_j = \max(y_{1,j}, y_{2,j}, \ldots, y_{L,j})$.

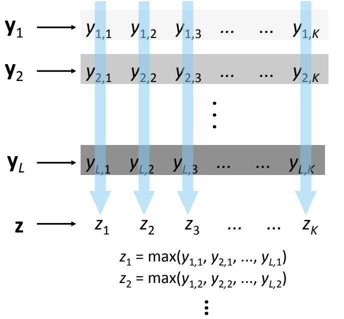
y_1 → y_{1,1} y_{1,2} y_{1,3} … … y_{1,K}
y_2 → y_{2,1} y_{2,2} y_{2,3} … … y_{2,K}
y_L → y_{L,1} y_{L,2} y_{L,3} … … y_{L,K}
z → z_1 z_2 z_3 … … z_K
z_1 = max(y_{1,1}, y_{2,1}, …, y_{L,1})
z_2 = max(y_{1,2}, y_{2,2}, …, y_{L,2})

Fig. 4. Max pooling of $L$ sparse codes

# IV. RF Signal Classification with Convolutional Sparse Coder

This section introduces a new method for RF signal classification based on feature extraction via sparse coding. We consider two case scenarios. In the first scenario, we consider that there is no labeled dataset for supervised classifier training. Here, we completely rely on unsupervised learning by sparse coding and dictionary training. The first scenario can be considered as blind source separation in the feature domain. In the second scenario, our approach is based on the semi-supervised learning framework.

## A. Sparse Coding Setup

Our view on sparse coding is that it is a customizable framework for feature extraction. The sparse coding setup in Fig. 1 is a realization based on matching or basis pursuits that emphasize reconstructive representation with the regularization on sparsity. For discriminative purposes, an OMP sparse coder evaluates the membership of a given input $\mathbf{x}$ to each dictionary atom with the inner product. Conceptually, this is equivalent to the way that K-means clustering evaluates the Euclidean distance between data and a cluster centroid, or that the

Gaussian mixture model computes the posterior probabilities given $\mathbf{x}$.

Since our eventual goal is classification, we want to optimally configure the sparse coding framework with the most suitable metric that examines the correlation of a received RF measurement to our dictionary atoms. We propose a convolutional sparse coder that maps an input vector $\mathbf{x} \in \mathbb{C}^N$ (samples of received signal) to the feature $\mathbf{y} \in \mathbb{R}^K$ with respect to the matched filter templates in dictionary $D$. The $i$th element in $\mathbf{y}$ is given by

$$y_i = \max |\mathbf{x} * \mathbf{d}_i| \,, \tag{6}$$

where $*$ denotes the convolution operator, and $\mathbf{d}_i$ the $i$th dictionary atom. We impose the similar regularization for $\mathbf{y}$, leaving only the $S$ largest values of $\mathbf{y}$ as are and setting the rest zeros.

The underlying principle behind our setup is matched filtering. Mathematically, the nonzero element in the convolutional sparse code $\mathbf{y}$ reflects the maximum correlation between the input $\mathbf{x}$ and the corresponding dictionary atom, which is some matched filter.

What are the matched filter templates that constitute our dictionary? Radio protocols employ specific pulse shaping to aid effective detection of known signals for the receiver. This pulse shaping function defines a matched filter template.

### B. Blind Signal Classification

For blind signal classification, we perform K-means clustering with sparse codes. This is essentially blind source (signal) separation performed in the feature domain.

### C. Semi-supervised Signal Classification

If both unlabeled and labeled datasets are available, we can use a semi-supervised method for signal classification. First, we perform unsupervised feature learning via sparse coding and dictionary training. Given the learned dictionary, we train linear 1-vs-all SVM classifiers. Assuming a multiclass classification problem with $M$ classes, each SVM is trained to classify signal class $j$ against class $k \neq j \; \forall j, k = 1, \ldots, M$.

At runtime, we take sample measurements, perform sparse coding and subsampling of sparse codes by max pooling, and predict the signal class label using the pooled sparse code.

## V. Improvement via Shift-invariant Sparse Coding

In the previous section, we have described a blind signal classification method based on sparse coding. We extend our baseline approach with shift-invariant sparse coding (SISC) that can compensate all possible shifts of a blindly sampled signal. SISC leads to more efficient representable basis functions that can be learned from unlabeled received signal samples. We argue that slightly high-level representations learned through SISC provide useful features to discriminate one RF signal to another.

### A. Preliminaries

Clustering time-series data is hard. If unable to address countless time shifts, it is well-known that the clusters extracted from (sub)sequences of a time series are close to random [23]. This makes unsupervised learning of efficient basis representations by the use of sparse coding fundamentally flawed. Shift-invariant sparse coding (SISC) [24], [25] is an extension of sparse coding that can accommodate possible time offsets of time-series data such as a radio signal. The SISC optimization problem is formulated as follows

$$\min_{\mathbf{y},\mathbf{d}} \sum_{i=1}^{n} \|\mathbf{x}^{(i)} - \sum_{j=1}^{K} \mathbf{d}_j * \mathbf{y}^{(i,j)}\|_2^2 + \lambda \sum_{i,j} \|\mathbf{y}^{(i,j)}\|_1$$
$$\text{s.t. } \|\mathbf{d}_j\|_2^2 = 1, \; 1 \geq j \leq K \tag{7}$$

where $\mathbf{y}^{(i,j)}$ represents the coefficient corresponding to the time-series input $\mathbf{x}^{(i)}$ (*i.e.*, received signal), the $j$th basis vector $\mathbf{d}^{(j)}$ from $\mathbf{D}$.

By expanding the convoluted summation term in the equation, we can treat the SISC problem as a massively large sparse coding problem. Unfortunately, doing so leads to computationally infeasible cases even for a moderate problem size. The existing algorithms [24]–[26] rely on heuristic solutions. We adopt the approach by Grosse *et al.* [27], which will be explained in detail.

### B. Efficient SISC Algorithm

In general, a sparse coding algorithm alternates between two convex optimization problems: 1) compute the sparse codes $\mathbf{y}$ by fixing the basis vectors $\mathbf{d}$ in dictionary $\mathbf{D}$ and 2) update the basis vectors by fixing the sparse codes. Key challenge to solve Eq. (7) is that each basis vector can appear in any possible shift, making every element in a basis vector contributed to many different terms in the objective function.

Grosse *et al.* [27] proposes an adept technique that transforms Eq. (7) into the frequency domain. Such transformation eliminates the intractable problem at the time domain and replaces with a new problem that is only mildly more difficult than classical sparse coding. Let us denote the discrete Fourier transform (DFT) of the basis vectors $\mathbf{D}' = (\mathbf{d}'_1, \ldots, \mathbf{d}'_K)$. Note that each $\mathbf{d}'_i$ is complex. Parseval's theorem implies that the DFT of $\mathbf{d}$ scales its $\ell_2$ norm by a constant factor, say $\alpha$. Also, the Fourier transform for a convolution of two vectors is the element-wise product of the Fourier transforms of the two. Therefore, we can reduce the optimization in Eq. (7) to

$$\min_{\mathbf{d}'} \sum_i \|\mathbf{x}'^{(i)} - \sum_j \mathbf{d}'_j \odot \mathbf{y}'^{(i,j)}\|_2^2 \;\; \text{s.t. } \|\mathbf{d}'_j\|_2^2 = \alpha \tag{8}$$

where $\mathbf{x}'$ and $\mathbf{y}'$ are the DFT of the input and its sparse code. The optimization is now over the vectors of complex-valued DFTs, and the Lagrangian can be decomposed as a sum of quadratic terms on each frequency component $k$

$$\mathcal{L}(\mathbf{d}', \boldsymbol{\gamma}) = \sum_k \left( \|\mathbf{x}'^{(i)}_k - \mathbf{Y}'_k \mathbf{d}'_k\|_2^2 + \mathbf{d}'^{*}_k \boldsymbol{\Gamma} \mathbf{d}'_k \right) - \alpha \mathbf{1}^\top \boldsymbol{\gamma} \tag{9}$$

with dual variables $\boldsymbol{\gamma} \in \mathbb{R}^K$. Note that $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ and $\mathbf{Y}'_k = \begin{bmatrix} \mathbf{y}'^{(1,1)}_k & \mathbf{y}'^{(1,2)}_k & \cdots \\ \mathbf{y}'^{(2,1)}_k & \mathbf{y}'^{(2,2)}_k & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$. The Lagrangian $\mathcal{L}(\mathbf{d}', \boldsymbol{\gamma})$ can be written as a function of only real variables using the real and imaginary parts of $\mathbf{d}'$. Furthermore, we can analytically compute $\mathbf{d}'_{\min} = \arg\min_{\mathbf{d}'} \mathcal{L}(\mathbf{d}', \boldsymbol{\gamma})$

$$\mathbf{d}'_{\min} = \left(\mathbf{Y}'^{*}_k \mathbf{Y}'_k + \boldsymbol{\Gamma}\right)^{-1} \mathbf{Y}'^{*}_k \mathbf{x}'_k \tag{10}$$

In this paper, we derive the dual optimization problem and optimize a solution over the dual variables $\boldsymbol{\gamma}$ via Newton's method. When the optimal $\boldsymbol{\gamma}$ is computed, the shift-invariant dictionary basis $\mathbf{d}'$ can be recovered by Eq. (10).

## VI. Evaluation

We evaluate the proposed blind and semi-supervised signal classification methods in MATLAB.

### A. Signals

We assume a mixture of known and unknown signals in an arbitrary radio channel. If the receiver detects an energy level above certain threshold, it samples the channel according to its bandwidth and stores the measurement for further processing, which will be explain in the next section. We have two classes of known signals, namely S1 and S2. By *known* signals, it means that we have the knowledge of pulse shaping applied during the baseband modulation and can perform matched filter detection. The known signals are:

- (S1) Single-carrier: QPSK with rectangular pulse;
- (S2) OFDM: QPSK modulated on-carriers with raised cosine pulse.

There are also two classes of unknown signals, S3 and S4, that we have no knowledge of their pulse shaping. The unknown signals have the following specifications.

- (S3) Single-carrier: QPSK with unknown custom pulse $p(t) = \frac{1}{2}[1 - \cos(\frac{2\pi t}{T_s})]$;
- (S4) OFDM: BPSK, QPSK, 16-QAM modulated on-carriers with $p(t)$.

### B. Experimental Methodology

*1) Generation and transmission of signals:* To generate signals, we have first generated random data bit stream $b_k$. The baseband signals are generated according to the following digital (I-Q) modulation schemes.

- BPSK: $d_{\mathrm{BPSK}}(t) = \sum_k b_k p(tkT_b)$
- QPSK: $d_{\mathrm{QPSK}}(t) = \sum_k b_{2k} p(tkT_s) + \sum_k b_{2k+1} p(t - kT_s)$
- 16-QAM: $d_{\mathrm{QAM}}(t) = \sum_k i_k p(tkT_s) + \sum_k q_k p(tkT_s)$
- OFDM: generated by `comm.OFDMModulator` method in MATLAB

For 16-QAM, $i_k$, $q_k$ are the in-phase and quadrature amplitudes, taking values ±1, ±3.

As mentioned earlier, we have used rectangular, raised cosine, square-root raised cosine, and custom pulse functions for baseband pulse shaping of the baseband modulated waveforms. The final carrier-modulated signal is given by

$$s(t) = A_c[d_I(t)\cos(2\pi f_c t) + d_Q(t)\sin(2\pi f_c t)],$$

where $f_c$ is the carrier frequency, and $A_c$ the carrier amplitude gain. The in-phase and quadrature components $d_I(t)$, $d_Q(t)$ are generated according to one of the I-Q modulation schemes above.

We transmit $s(t)$ through the AWGN channel at 20 dB and 0 dB SNR. Hence, the measurement at a receiver constitutes noisy samples. For every class, we generate two datasets. There are 1,000 signal examples per each dataset. We use the first dataset for training, and the other for evaluating classification performances.

*2) Data processing and classification pipeline:* The data processing and classification pipeline is depicted in Fig. 5. The measured signal samples are vectorized to patches of size $N = 64$. Note that an I-Q modulated signal is complex-valued, hence the received samples are also complex, *i.e.*, $\mathbf{x} \in \mathbb{C}^N$. We can train the dictionary using the received samples via unsupervised K-SVD algorithm (without knowing what their classes are). However, we take the following generative approach for dictionary $D$.

1) $D$ has $K = 100$ dictionary atoms
2) Each atom $\mathbf{d}_i$ has matching size $N = 64$ and is complex-valued
3) $D$ is divided to 4 regions—first 20 atoms belong to the family of rectangular pulses, second 20 atoms to raised cosine family, third 20 atoms to square-root raised cosine; the last 40 atoms are randomly generated

We use a convolutional sparse coder whose operation is described by Eq. (6) given a patch $\mathbf{x}$ of the received signal samples. Sparse code $\mathbf{y}$ has dimension $K$ and is real-valued ($\mathbb{R}^K$). We use the max pooling factor $M = 4$. Note that the pooled feature vector $\mathbf{z}$ has the same dimension as $\mathbf{y}$ and is also real.

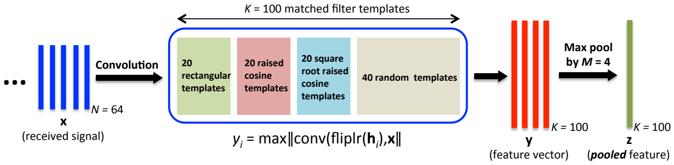

Fig. 5. Sparse coding setup with convolution for RF signal classification

In summary, the feature transformation $\mathbf{x} \mapsto \mathbf{y} \mapsto \mathbf{z}$ takes place by sparse coding of sequentially-fed raw input patches followed by max pooling. The pooled feature vector $\mathbf{z}$ is used for classification.

*3) SVM classifier training:* Fig. 6 explains 1-vs-all SVM training. We have trained two linear SVM classifiers using the pooled feature vectors $\mathbf{z}$. The first SVM classifies the signal class S1 against all others. For this, we prepare labeled datasets $\{\mathbf{z}^{(j)}_{S1}, +1\}^{T}_{j=1}$ and $\{\mathbf{z}^{(j)}_{S2\cup S3\cup S4}, -1\}^{T'}_{j=1}$. Similarly, the second SVM that classifies S2 against all others are trained with labeled datasets $\{\mathbf{z}^{(j)}_{S2}, +1\}^{T}_{j=1}$ and $\{\mathbf{z}^{(j)}_{S1\cup S3\cup S4}, -1\}^{T'}_{j=1}$.

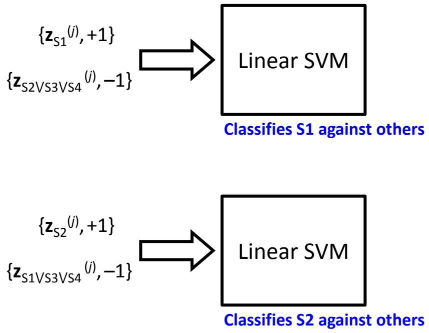

Fig. 6. 1-vs-all SVM training for semi-supervised approach

*4) Evaluation metric:* We compute recall or true positive rate (TPR) and false alarm rate (FPR) to evaluate classification accuracy.

$$\text{Recall} = \frac{\sum \text{True positives}}{\sum \text{True positives} + \sum \text{False negatives}}$$

$$\text{False alarm} = \frac{\sum \text{False positives}}{\sum \text{False positives} + \sum \text{True negatives}}$$

### *C. Results and Discussion*

For blind classification, we try to see whether or not K-means clustering in the pooled feature ($\mathbf{z}$) domain gives natural separation. Setting the number of clusters $K = 4$ for K-means (not to be confused with the number of atoms $K$ in dictionary $D$), we have been able to see the separation that we seek for. By counting the cluster majority signal class and minorities, we have computed the recall and false alarms. Table I summarizes the accuracy of the blind signal classification via K-means.

The blind classification performance is reasonably good considering that we do not use any prior knowledge about these signals, yet we can achieve up to a 70.3% recall at 24.6% false alarm at SNR = 10 dB. If a labeled dataset were available for semi-supervised training, our classifiers would have achieved a 87.8% recall with 14.1% false alarms, all without any protocol-specific knowledge about modulation of radio signals. In Fig. 7, we present the confusion matrix for 1-vs-all SVM classifier trained using signal sampled received at SNR = 10 dB. Similarly, the confusion matrix for SMV classifier trained under SNR = 0 dB is presented in Fig. 8.

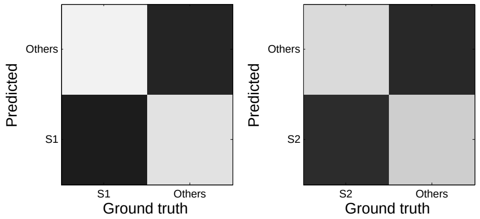

Fig. 7. Confusion matrix for SNR = 20 dB (darkest box: 0.89, lightest: 0.06)

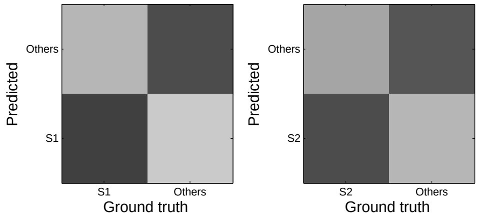

Fig. 8. Confusion matrix for SNR = 0 dB (darkest box: 0.73, lightest: 0.22)

TABLE I
CLASSIFICATION ACCURACY (VALUES IN PARENTHESES ARE FOR SNR = 0 dB)

| | Recall | False alarm |
|---|---|---|
| Scenario 1<br>(Blind classification) | 0.703 (0.582) | 0.246 (0.367) |
| Scenario 2<br>(Semi-supervised) | 0.878 (0.726) | 0.141 (0.262) |

## VII. RELATED WORK

Our signal classification methods are inspired by the way that sparse representations of raw image, audio, and text data are used in computer vision and pattern recognition. Wright *et al.* [28] have developed a recognition system that can classify an image of human face using sparse representations of image segments, which is a similar idea to ours. Pooling multiple sparse features to make an aggregate representation is widely studied in computer vision. The original idea of spatial pooling techniques dates back to Riesenhuber and Poggio [29]. Heisele, Ho, and Poggio [30] explain useful techniques of applying SVM for multi-class classification such as training a 1-vs-all classifier in our semi-supervised approach.

## VIII. CONCLUSION

We have introduced a blind signal classification method based on sparse coding. Our method is motivated by an active area of research in sparse representation learning [31]. With no prior knowledge or assumptions on a blindly sampled signal, we take advantage of correlating it to an overcomplete dictionary of known matched filter templates, which can be pregenerated or trained by an unsupervised learning algorithm. This coding process yields a discriminative feature that captures the variability of correlations measured by convolving the signal with respect to each dictionary atom.

As our goal is to build a discriminative framework for classification tasks, we have exploited sparsity by regularizing over the convolutional filter outputs and leaving only the largest several values as are. The empirical results for classification are promising. We have designed a simulated experiment for blind classification similar to blind source separation and found that our method can achieve up to a 70.3% recall at 24.6% false alarm rate at a reasonable SNR of 10 dB without any protocol-specific knowledge about simulated radio signals. If a labeled dataset were available for supervised training, our classifiers would have achieved a 87.8% recall with 14.1% false alarm.

## References

[1] Q. Zhao and B. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing Magazine*, May 2007.

[2] B. Wang, Y. Wu, K. Liu, and T. Clancy, "An Anti-jamming Stochastic Game for Cognitive Radio Networks," *IEEE JSAC*, vol. 29, no. 4, 2011.

[3] H. Li and Z. Han, "Dogfight in Spectrum: Combating Primary User Emulation Attacks in Cognitive Radio Systems, Part I," *IEEE Trans. on Wireless Communications*, vol. 9, no. 11, pp. 3566–3577, 2010.

[4] S. Sodagari and T. Clancy, "An Anti-jamming Strategy for Channel Access in Cognitive Radio Networks," in *Decision and Game Theory for Security*. Springer LNCS, 2011, vol. 7037, pp. 34–43.

[5] Y. Gwon, S. Dastangoo, C. Fossa, and H. Kung, "Competing Mobile Network Game: Embracing Antijamming and Jamming Strategies with Reinforcement Learning," in *IEEE Conference on Communications and Network Security (CNS)*, 2013.

[6] K. Liu and Q. Zhao, "Distributed Learning in Multi-Armed Bandit With Multiple Players," *IEEE Trans. on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, Nov 2010.

[7] L. Lai, H. Jiang, and H. Poor, "Medium Access in Cognitive Radio Networks: A Competitive Multi-armed Bandit Framework," in *Asilomar Conference on Signals, Systems and Computers*, 2008.

[8] Y. Gwon, S. Dastangoo, and H. Kung, "Optimizing Media Access Strategy for Competing Cognitive Radio Networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2013.

[9] B. Olshausen and D. Field, "Emergence of Simple-cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[10] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[12] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Dictionary Learning for Sparse Coding," in *Proc. of International Conference on Machine Learning (ICML)*, 2009.

[14] S. Chen, S. A. Billings, and W. Luo, "Orthogonal Least Squares Methods and Their Application to Nonlinear System Identification," *International Journal of Control*, vol. 50, pp. 1873–1896, 1989.

[15] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[16] L. Bottou, "Online Learning and Stochastic Approximations," in *Online Learning in Neural Networks*, 1998, pp. 9–42.

[17] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 11, 2006.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[19] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[20] L. M. Manevitz, M. Yousef, N. Cristianini, J. Shawe-taylor, and B. Williamson, "One-Class SVMs for Document Classification," *Journal of Machine Learning Research*, vol. 2, pp. 139–154, 2001.

[21] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[22] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A Theoretical Analysis of Feature Pooling in Visual Recognition," in *ICML*, 2010.

[23] E. Keogh and J. Lin, "Clustering of Time-series Subsequences is Meaningless: Implications for Previous and Future Research," *Knowledge and Information Systems*, vol. 8, no. 2, 2005.

[24] M. S. Lewicki and T. J. Sejnowski, "Coding Time-Varying Signals Using Sparse, Shift-Invariant Representations," in *NIPS*, 1999.

[25] E. Smith and M. S. Lewicki, "Efficient Coding of Time-Relative Structure Using Spikes," *Neural Computation*, vol. 17, no. 1, 2005.

[26] T. Blumensath and M. Davies, "Sparse and Shift-Invariant Representations of Music," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 1, 2006.

[27] H. Grosse, R. Kwong, and A. Y. Ng, "Shift-invariant Sparse Coding for Audio Classification," in *Proc. of Uncertainty in Artificial Intelligence (UAI)*, 2007.

[28] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. M., "Robust Face Recognition via Sparse Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[29] M. Riesenhuber and T. Poggio, "Hierarchical Models of Object Recognition in Cortex," *Nature*, vol. 2, no. 11, pp. 1019–1025, 1999.

[30] B. Heisele, P. Ho, and T. Poggio, "Face Recognition with Support Vector Machines: Global versus Component-based Approach," in *ICCV*, 2001.

[31] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.